Chapter 12
Operations Architecture

OVERVIEW: DEFINITION OF OPERATIONS ARCHITECTURE

Although the visibility of information technology within a business is typically driven by the Execution and Development Architectures, the Operations Architecture plays a crucial but often underestimated role in the successful delivery of computing service to end users.

An *operations* architecture describes the tools and support services required to keep a production system up and running well. It differs from an execution architecture and development architecture in that its primary users are systems administrators and production support personnel. In addition, it differs from the operations *infrastructure* in that the operations infrastructure represents operations processes and organization as well as the technologies and tools.

This chapter will describe several categories of Operations Architecture tools. It will also discuss netcentric computing's impact on operations architecture tools and technologies.

When considering operations, addressing the operations tools alone would be misleading. To be successful, the operations tools must integrate closely with an effective organizational structure as well as a set of operations processes driven by the requirements of the user community. Although this chapter focuses only on the operations architecture and tools, a framework called MODE, or Management of Distributed Environments, addresses the process and organization as well as the technology facets of operations. MODE is described in detail in Section III of this book.

EVOLUTION OF THE OPERATIONS ARCHITECTURE

In the mainframe environment, operations tasks are performed by those in the data center who constantly watch, monitor, and react to problems with the host or network.

Keeping a mission-critical client/server application system available and under control, while providing a high level of service to the end user,

is more complex and difficult than in a mainframe environment. Unfortunately, not all organizations are aware of this complexity as they should be.

When client/server computing first emerged, organizations expected the cost and complexity of operations to be reduced because of reduced administration and because of common operating systems on workstations and servers. Time has shown that client/server environments tend instead to add rather than reduce complexity, therefore increasing operations costs.

More recently, netcentric computing has emerged as the next technology generation that will coexist with host and client/server environments. Again, while the initial hype around netcentric suggested that it would significantly simplify operations, experience is beginning to indicate that netcentric only adds an additional level of complexity through additional processes, tools, and support services, thus creating an environment even more potentially difficult and expensive to manage. The operations architecture now needs not only to keep an organization's internal production systems up and running but also to maintain production systems that extend to business partners and customers.

The complexity and cost of operations architecture keeps increasing, which suggests a strong need for a structured and disciplined approach to implementation of tools and technologies to support eased operations.

OPERATIONS ARCHITECTURE TOOLS

When implementing an operations architecture, an organization must select among the wide variety of tool categories. Tool categories cover the spectrum of functions provided by the operations organizations, from software distribution to Help Desk. Although the industry has slowly progressed toward the vision of a single, consolidated multifunction operations product, usually a suite of products must be purchased and therefore integration work must be performed.

The most common categories of operations tools support such things as

- Software distribution
- Configuration and asset management
- Fault management and recovery management
- Capacity planning
- Performance management
- License management
- Remote management
- Event management
- Monitoring and tuning
- Security

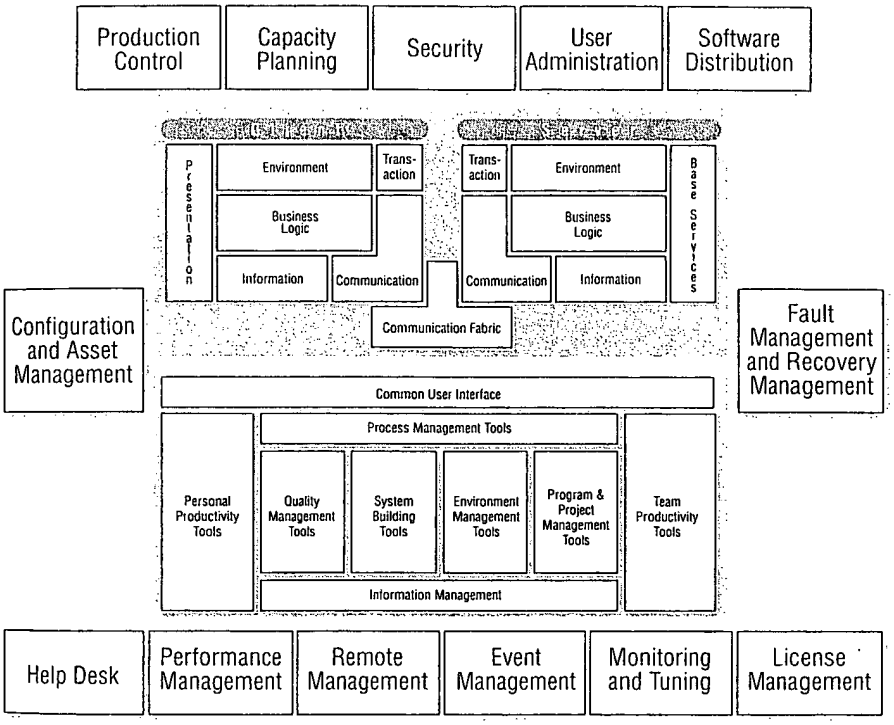

Exhibit 1. Operations Architecture Framework.

- User administration
- Production control
- Help desk

These tools must provide visibility and control over the events of a distributed environment. In essence, they can be viewed as support and control mechanisms for both the execution and development architectures. This relationship is shown in Exhibit 1, where the major categories of operations tools are depicted as supporting the development and netcentric execution architectures.

An overview of each of these tool categories, as well as some implementation considerations for each, are provided through the remainder of this section.

SOFTWARE DISTRIBUTION

Software distribution is the automated delivery to, and installation of, applications and systems software on servers and end user devices (e.g., workstations, kiosks, etc.). This can be for an organization's internal computing

environment, as well as for its extended one, i.e., its business partners and customers. The architectural support required to support software distribution is largely driven by the numbers of workstations, servers, and geographic locations to be served.

For a relatively small network of workstations in a single physical location — where it is not anticipated that software changes will be frequent — a manual approach should not be automatically ruled out. A manual approach involves systems management personnel loading software upgrades on each workstation or server by physically visiting each machine. This approach does not scale well, however, when either large numbers of workstations or servers in a single environment need to be updated or multiple geographic locations are involved.

When it is unrealistic to use a manual approach, an organization should consider adding automated software distribution tools to the operations architecture. Many products from leading vendors such as Microsoft, Tivoli, and Hewlett-Packard are on the market today that include or specialize in automated software distribution. Systems developers must look for several important features, depending on the specific support requirements.

Creating a Software "Distribution"

The server component of a software distribution solution enables administrators to build distribution packages and to control distribution. A distribution is a package of related software files, data, and installation scripts that form an installable unit.

Few significant application installations, systems software installations, or even upgrades can be achieved simply by sending a single file. Configuration files (e.g., config.sys) and system files (e.g., autoexec.bat and .login) as well as multiple software files for a particular application or systems software component, often require changes.

In addition, it is usually desirable to upgrade multiple applications or combinations of systems software and applications in a single distribution rather than performing multiple independent software distributions. Bundling software upgrades together also reduces the amount of release testing required.

A distribution is created by selecting the files and scripts, often through a point-and-click interface. The components are then combined into a single file for transmission. Some software distribution tools provide compression capabilities to reduce the physical size of the distribution. This is particularly important in a wide area network (WAN) environment where line speeds are an issue.

Scheduling a Distribution: Push vs. Pull

There are multiple approaches to scheduling software distributions. Some solutions use a rigid scheduling mechanism that requires all target machines to be powered on at a specified time when the software distribution is to occur. This mechanism could be characterized as a "push" strategy, where the server machine pushes the software to the client machines at a specified time.

The push strategy may be possible in some smaller situations, but in large organizations it is difficult to ensure that users will leave their machines on, particularly if it is common practice to turn them off at the end of the day.

A more flexible approach is the "pull" strategy, where the workstations check for software updates and pull the software from the designated server or servers at log-in time. Thus, when the user signs on either in the morning or at some point during the day, any pending updates are downloaded to the client machine. When combined with a forced log-off capability, which most networks support, this can effectively mimic the push strategy without the attending problem of some machines being powered off.

Neither the push nor pull scheduling approach is sufficient when large numbers of target workstations are involved. For example, a sales office automation system developed several years ago and used by 1,400 salespeople distributed across scores of locations encountered a problem with these strategies on its first major software upgrade. The sales office used the pull strategy because it was not feasible to have all workstations, locations, and dial-up users connected and powered up at the same time. The distribution was scheduled to be available when the users logged in on Monday morning. This was a substantial functional upgrade to the system, so the software distribution was several megabytes in size.

The problem was that 1,400 machines could not simultaneously download one copy of software off of a server. As a result, most users were unable to retrieve the new software or use the system for several days. The problem was eventually solved by "staging."

Software Distribution Staging

Faced with the problem of scale, two alternatives can be considered. One is simply to acquire more servers with more copies of the software to be distributed. Of course, this is an expensive solution, particularly when these machines are not needed for any other purpose.

An alternative solution that may be better involves the concept of staging. Software distribution staging works by sending a new version of the software in advance of the cut-over date. In effect, the client machines have

two versions of the application physically resident simultaneously, but only one is in use.

The existing software is used until the present cut-over date is reached. At that time, the client portion of the software distribution architecture automatically completes the installation and redirects the user to the new version. Using this approach, it is possible to selectively download the software update to subsets of machines well in advance of the cut-over date, thus eliminating the bottleneck.

An enhancement of staging is the ability to cut over to the new version on the receipt of a small command file rather than a preset date. This gives operations more flexibility to alter the cut-over date due to unanticipated events. For example, many adopters fail to anticipate the requirements of having multiple copies of applications stored simultaneously when determining the size of the workstation hard disks required for the users.

Remote Installation

Most software distribution solutions include a client portion as well as a server that resides on the target machine. The client software is responsible for installation of the software distribution onto the target machine's hard disk.

The first step is the unbundling (and uncompressing) of the distribution into the component files, data sets, and scripts (although the better products will first check to see that the required disk space is in fact available). Next, any preinstallation scripts are executed. These scripts may do such various tasks as checking for required components or adding or modifying lines in the target machine configuration or systems files that will be required by the new software (e.g., changing the number of buffers or adding a line to install a necessary driver at startup time). The directories in which the software is to reside are checked or created, and then the actual software files are moved into the proper location on the hard disk. At this point a postinstallation script may be invoked that could include rebooting the machine so that the changes to the system and configuration files can take effect.

Cascaded Distribution

In large networks, where tens or even hundreds of servers support individual groups of workstations, a "cascaded" approach may be required. A cascaded software distribution approach allows for a central administrator to schedule the distribution of software updates to designated servers within the network environment. These servers, in turn, distribute the software updates to their associated client workstations.

This approach allows the simple push and pull strategies to be used for larger numbers of workstations without requiring staging. It also better utilizes the servers and communications links in these larger environments. Most products that support a cascaded approach also support staging concepts as well, thus providing much flexibility in how software is to be distributed.

Relationship to Configuration Management

Many of the available software distribution packages offer integrated asset and configuration management capabilities (described in the next section) as well. Although not specifically required for software distribution, these functions are naturally related, and integrating these capabilities simplifies the operations architecture.

A useful feature is the ability to check to see whether all the system and application files required by a software distribution, but expected to be already resident on the target machines, are in fact there. For example, when sending a Visual Basic application, this feature checks the target machine to see that the user has not moved or deleted a required file such as VBRUN001.DLL.

A full-function software distribution system needs many of the same capabilities as a configuration management or asset inventory tool. The trend toward combining these functions within the products market will certainly continue.

Error Handling Reporting

When dealing with larger networks of workstations, errors inevitably occur in the software distribution process. There may be insufficient disk space or a required component may be missing. Capability is required both to report errors and to take appropriate actions.

Error reporting normally takes the form of a distribution log file that records success, failure, or errors encountered. In some cases a more active form of error reporting is required, where e-mail messages may be automatically generated and sent to either the administrator or, in some cases, the affected user. If a fatal error is detected, the software distribution system should be capable of reversing any changes made to that point and restoring the user's machine to its previous state.

Platform Constraints

The choice of software distribution tools is somewhat limited by the types of workstations, servers, operating systems, and networking software in use. Some products are UNIX based and support only UNIX clients or at least require UNIX servers. Others work well with Windows workstations.

In environments where intermittently connected dial-up users need to be provided with software distributions, the existence and unreliability of dial-up connections adds more complexity to the software distribution task.

CONFIGURATION AND ASSET MANAGEMENT

To manage a netcentric environment successfully, one must have a solid understanding of what is where, and one must maintain rigor in the change control procedures that govern modifications to the environment. Configuration and asset management information that may need to be tracked includes such details as product licensing information, warranty information, vendor names, logical and physical device information (such as total capacity and current utilization), product configuration tracking, software and data version levels, network configuration parameters, physical location, and perhaps accounting information.

For relatively small netcentric environments — under 100 workstations, for example — it may be reasonable to use a manual approach. A manual approach keeps track of information in a personal computer database or in a collection of spreadsheets. For larger environments the manual approach has proven time and again to be inadequate, and automated tools are required for collecting asset and configuration information and for periodically auditing the environment.

In larger netcentric environments, it is often necessary to have an underlying configuration and asset management database or repository. This database becomes a key information source for those managing, maintaining, and adding to the environment. However, it is only useful if the database is current, reliable, and perceived to be that way. Otherwise, configuration and asset management databases quickly fall into disuse.

Automated Tools

Automatic asset and configuration collection capability is included in many vendor solutions, including OpenView from Hewlett-Packard (HP) and POLYCENTER Systems Census from Digital Equipment Corp. These products can interrogate the network, discover network and computing devices, and collect related information. In addition, these products can perform the needed periodic auditing to detect changes to the environment over time, for example, when a user moves a machine or installs a network game.

Another important and related feature is the ability to restore a machine to a known or initial configuration for problem resolution. The configuration and asset management architecture component both provides facilities for determining the correct initial state for a given machine or network

device and initiates any software distribution or configuration changes needed to bring the device back within compliance.

For more dynamic environments, where machine and network configurations are changing frequently, it is even more important to have an active configuration and asset management system. The capability to automatically change configurations of a large number of machines and network components or even to roll back to previous configuration settings for any particular device becomes increasingly important.

Many products that can form the core of asset and configuration management are bundled with additional related functions for fault and performance management. HP's OpenView is just one example of an integrated suite of operations architecture products that can greatly simplify piecing together an integrated architecture.

Multivendor Problem

When sourcing asset and configuration management products from the marketplace, it is important to consider that they are quite particular in the types of networks and devices they can support. For example, the field of suitable asset and configuration management products becomes quite limited when the netcentric components are not in the "mainstream" — such as the Pick operating system or Wang servers — although management standards such as the Simple Network Management Protocol (SNMP) have increased the coverage of many solutions.

Second, products that specialize in serving smaller market segments, such as Macintosh clients or lesser known network protocols, sometimes do not support as wide a variety of client machines, operating systems, mainframes, and network protocols.

Finally, integrating multiple management platforms is complex, costly, and in many cases impractical.

In sum, if the hardware, systems software, and networking that make up the environment are out of the business computing mainstream, it is more difficult to find adequate configuration management solutions from the marketplace. This leaves developers with the daunting challenge of custom development of configuration and asset management capabilities.

Impact Analysis

A well-functioning configuration and asset management component becomes a vital information source for conducting impact analysis for any requested changes to the environment. The frequency with which unexpected negative side effects are caused by relatively minor configuration

changes to the netcentric environment has been an embarrassing and frustrating surprise for many adopters of the technology.

Much of the source of these problems relates to the high number of execution architecture components and complex interdependencies between them. Another problem is the reality that most netcentric networks involve numerous independent vendors. Changing even the release level of one systems software component may have a ripple effect and may require updates to, or newer versions of, additional software components or applications.

To support this type of impact analysis, dependency information must be maintained. For example, version X of the Oracle database management system requires version Y or greater of the HP-UX operating system and version Z of yet another vendor's Transmission Control Protocol/Internet Protocol product.

It is not uncommon for a user organization to wish to return to a previous operating system release to acquire an application package that does not yet support the latest operating system version. Without an effective configuration and asset management system that maintains relationship information, it is purely guesswork if in fact the proposed version change will break any required dependencies. Unfortunately, this is how many organizations approach this problem in the netcentric world today — typically with unsatisfactory results.

Appropriate Degree of Standardization

One of the keys to effective configuration and asset management is enforcing the appropriate degree of standardization across environments. For large netcentric networks, where thousands of workstations are involved, it is not feasible to effectively manage the environment if each machine has its own unique configuration and combination of software products. On the other hand, it is not typically appropriate to give thousands of users the exact same configuration if the users perform different functions within the organization.

For example, users in such diverse areas as sales, product development, and human resources are likely to require different computing capabilities. The goal is to strike the correct balance between standardization, which simplifies the required operations architecture and tasks, and accommodation to each business area's unique computing needs.

FAULT MANAGEMENT AND RECOVERY MANAGEMENT

Failure control is important in a netcentric environment. The presence of heterogeneous equipment, however, makes it difficult to determine the origins of

a fault. Multiple messages may be generated within the system from a single fault, making it difficult to separate the fault's cause from its effects.

The fault management services of an operations architecture assist in the diagnosis and correction of system faults. Faults may include network-, server-, workstation-, or even application-level faults. Fault diagnosis may require services for isolation, viewing of host, server, and workstation error logs; and determining the software and data versions and configurations of affected machines.

Managing Networks

Fault management services also encompass network management and diagnostic tools for monitoring and reporting on network traffic and failures. Additional diagnostic tools such as protocol analyzers are required in some cases to determine the true source of the problem.

A wide variety of tools and products for fault management is available on the marketplace. When selecting a tool or vendor, it is important to take into consideration the breadth of netcentric networking components to be managed to ensure that the fault management products selected have the necessary breadth of vendor coverage.

Another factor to consider in this selection is the choice between integrated operations environments (typified by HP's OpenView or CA-Unicenter TNG), and point solutions that provide only one function. Although most integrated tool sets today do not adequately address the full breadth of fault management and diagnostic requirements, they can reduce the number of vendors and the complexity of integrating these point solutions.

Once again, multivendor environments increase the complexity and difficulty of providing fault management services. It may be difficult or even impossible to find products that cover the scope of capability required as well as the various hardware and systems software components needing to be managed. In larger netcentric installations, some level of centralized fault management is usually employed to leverage specialized skills.

Recovery capabilities are also included in failure control. Recovery capabilities span the range from those required to bring up a device after it has failed to those required in the event of a major disaster. With critical business applications being rolled out on distributed technologies, the recovery of these systems must be easy, quick, and efficient. Loss of the system for even a short period can result in significant financial losses to the business.

A wide variety of architectural services may be required for fault recovery. These range from strictly network-oriented components (for restoring

links or reconfiguring components) to more systems-level components (for restarting processes on machines or restoring databases). More involved tasks, such as the distribution of software fixes to workstations or servers, may require the ability to remotely reboot and reinitialize machines, printers, or other network components.

CAPACITY PLANNING

Capacity planning tools focus on components of an environment such as the network, physical space, and processing power to understand the need to change the capacity of those components based on organizational changes. The tools typically focus on components that are considered to be heavily sensitive to changes in computing resource usage. The tools may use historical management data combined with estimates for growth or changes to configuration to simulate the ability of different system configurations to meet capacity needs.

Capacity Planning tools can sometimes be integrated into a larger integration platform, or they can be standalone applications.

PERFORMANCE MANAGEMENT

Performance management is more difficult because of the lack of tools to assist with performance in heterogeneous environments. Performance is no longer confined to the network or to the central processing unit. Performance needs to be viewed in an end-to-end manner, accounting for all the factors that affect the system's performance relative to a user request.

The creation of a customer order, for instance, may involve multiple server accesses for data and information to be exchanged between the workstation and the host. The performance relative to the entire business event needs to be considered, not simply the performance of a single component involved. To make performance management even more difficult, not all devices provide performance information. It may be necessary to develop surrounding processes that monitor the performance of devices to calculate and provide end-to-end performance information.

LICENSE MANAGEMENT

Since the advent of computer networks that allow applications software to be shipped around the network as required, the issue of license management has become increasingly important. Applications software vendors have been experimenting with various licensing strategies, including unrestricted site licenses, fixed concurrent user licenses, and floating licenses that actually enforce the restriction on concurrent users.

Independent of these actions by software vendors, large organizations have been struggling to keep a handle on exactly what software products they own and how many copies they own. They have also been working to ensure that they are in compliance with software licensing agreements while not paying for more copies of software than they truly need.

In netcentric environments, license management is challenged by other issues such as the unpredictability of number of copies required, and the management of licenses distributed to anonymous users on the Internet.

The market for license management solutions is immature at this time. The problem is difficult to solve in the absence of standards to which applications software vendors can adhere. From an operations perspective, however, the risk is that major applications software vendors will thrust their own license management solutions upon their customers, leaving the operations organization no choice but to support multiple and nonintegrated license management solutions. The problem becomes even more complex as vendors move to more of a usage-based charge, requiring that billing information be extracted from the license management component of the operations architecture.

In addition to guaranteeing compliance with software licensing agreements, license management provides valuable information about which people and how many people are actually using a given software product. If, in fact, usage statistics indicate that the organization has overpurchased, it may be possible to realize some savings by reducing software licensing agreements.

REMOTE MANAGEMENT

As distributed environments allow users more flexibility in terms of where they work, the ability of a centralized support group to effectively manage remote users is challenged. Visibility to a user's system configuration is only possible by physically sitting at the workstation and diagnosing problems or by accomplishing the same remotely.

Remote Management tools allow support personnel to "control" a user's desktop over a network so that they do not need to be physically present at a workstation to diagnose problems. Once control of the desktop is established, screen updates for the controlled desktop are displayed at both locations. The support person is then effectively sitting at the workstation he/she controls and can do necessary diagnostics.

In addition to problem diagnosis, remote management tools can provide visual explanations to user questions. For example, if a user has a question about a certain application feature, the support person may remotely control

the user's desktop and then walk through the solution while actions are displayed on the user's screen.

Remote Management tools are also useful in organizations where 24x7 support is required. Rather than requiring support personnel to be physically present for all events, they may be able to dial in through remote management tools from home and accomplish the same tasks. The ability to perform these tasks remotely can have positive effects on overall support costs through a reduction in the amount of time needed to resolve problems.

Remote Management products may come bundled with an integration platform such as HP OpenView or Tivoli TME, or they may be purchased as third-party software packages.

EVENT MANAGEMENT

In addition to hardware devices, applications and systems software also generates events. Common event-handling mechanisms are required to provide information to management in a simple, consistent format and to forward on important events for management purposes.

In most environments, events should follow an open format rather than a proprietary one as managed devices are rarely all from a single vendor. Filtering capabilities may also be needed at remote locations to prevent the streaming of events to central/master management consoles.

MONITORING AND TUNING

The number of devices and the geographic disparity of devices in a netcentric environment increase the effort required to monitor the system. The number of events generated in the system rises due to the increased complexity. Devices such as client machines, network components, and servers generate events on startup or failure to periodically report device status.

SECURITY

The security concerns of netcentric environments have been widely publicized. Although requirements for netcentric security architectures are constantly evolving as new security breaches are discovered, there are many tools categories that can help provide reasonable levels of security.

It is a common misperception, however, that security technologies in and of themselves provide the necessary protection from intrusion. It is equally important to have in place the people and processes to detect and react to security events. Without these components, the generation of

security information by technologies can only go so far in protecting the assets of an organization.

Because of the priority and complexity of the security subject, a separate chapter in this book has been devoted to the subject. Refer to Chapter 28 for a detailed discussion of the people, process, and technology aspects of security.

USER ADMINISTRATION

The netcentric environment introduces many new challenges to the task of user administration. The majority of these stem once again from the dramatically increased number of system components. Adding a user to the system may require adding a user to the network, one or more server operating systems, one or more database systems (so that the user can access data), an e-mail system, and an existing host-based system.

In some cases, the addition of a user has required entries to be added to upward of 15 individual system components. Even determining all the subsystems to which a user must be added can be a frustrating and often unfortunately iterative task with the user.

Deleting a user from the system is even more difficult. Unless careful records are kept, it can be very difficult to determine to which machines, databases, and applications the user had been added originally so that this information can be deleted. From an administration standpoint this may seem to be only a headache, but from a security standpoint it represents a substantial risk.

Problems related to adding or deleting users from a system are exacerbated by the number of user types and the dissimilarity of their configurations. For example, in a financial services organization, mortgage officers, commercial lending officers, and risk management users all have access to different combinations of systems and servers. If one wants to alter or delete access for one particular user, it may be necessary to know something about that person's role in the organization in order to determine which components that user was added to. The problem becomes completely unmanageable as individual users within a department or work group themselves have unique access privileges.

In larger netcentric environments with many components and combinations of user capabilities, user administration becomes a significantly more resource-intensive task than in the centralized mainframe environment. In the mainframe world, it was possible to acquire tools such as RACF that could interface with the various systems software components to add, change, or delete user attributes. It was possible to develop these products largely because of the homogeneous and consistent nature of the mainframe

environment. (Typically, all the systems software were sourced from one vendor, such as IBM or Digital Equipment Corp.)

In the more heterogeneous netcentric environment, few tools can manage user administration across a broad variety of products. For example, adding a user to the Sybase database product is different from the Informix product or Oracle product, and few user administration solutions cover all the combinations of even a typical netcentric environment. The result is often that operations must train personnel in how to do user administration for the various systems software products within the environment and must develop custom utilities and tools that are unique to the shop for automating user administration tasks.

Most user administration products on the market today focus on the operating system aspect of the problem (adding user access to the server, setting file permissions, and group associations). Although these solutions are certainly helpful, they do not cover many of the more difficult user administration challenges such as database access, e-mail, and networking software. Each of these products often comes with its own administration tools which may simplify the individual administration tasks but do little to help with providing an integrated user administration approach.

An alternative approach to user administration is to implement a Single Sign-On (SSO) application. These applications are meant to eliminate the need for users to remember user names and passwords to all of their business applications. The first time they log in, users enters a user name and password into the SSO application, which then automatically logs into applications through a scripting process. An additional advantage to this approach is that through implementing SSO, a database that maps users to the applications they access is created. This significantly simplifies user administration, and can increase security as well. A key drawback to SSO applications is failover. If a SSO server fails, users cannot access applications as they do not remember passwords to all their applications.

PRODUCTION CONTROL

In distributed environments, processes may be taking place across the entire system on multiple platforms in either a parallel or a serial fashion. Batch dependencies may be required across platforms, and multiple time zones may be involved.

In addition, many non-mainframe-based products do not provide production scheduling capabilities included with the platform. For these reasons, scheduling processes across a distributed environment can be quite complex, requiring significant management effort to ensure that the processes run smoothly. Many other day-to-day activities become more difficult in a distributed environment, including print management, file transfer

and control, mass storage management, backup and restore, archiving, and system startup and shutdown.

Backup and Restore/Archiving

Backup and restoration processes become more complex in a distributed environment as business-critical information becomes distributed across the system. Backup strategies must coordinate the information across the system and must determine where the backup copy or copies of information will reside.

As with centralized computing environments, restoration processes are directly dependent on how backup was performed. A single restore process no longer suffices. Depending on a particular fault, restoration services may only need to be performed for a portion of the system, while the rest of the system stays up and running.

Some technical expertise may be required on site to perform backups/restores (e.g., on/from server tape drives). In this case, backups and restores may need to take place during the business day, potentially affecting the processing that takes place at the distributed sites. If coordination of the distributed and centralized backup/restore strategies requires participation from someone at the remote locations, scheduling of these tasks becomes more difficult and complex, particularly across time zones.

The issues surrounding archiving are quite similar to those surrounding backup. Distributed architectures also place limitations on the amount of information that may be archived on a remote system as a result of the space limitations on servers and workstations.

Additional problems are created with archiving in a distributed environment because users have no incentives to perform housekeeping tasks on their devices. Depending on the users' ability to store information on their machines or on the local server, these machines may become cluttered with seldom-used files. Lack of space may affect other processes that need to take place on these devices, such as software and data distribution.

HELP DESK

As netcentric computing puts the operations Help Desk closer to the "end user" in terms of visibility and influence, the Help Desk will need to become integrated with the business processes being supported through netcentric. Unless the operations Help Desk is well integrated with the business process, there is risk that the user may be given information that is incorrect, forwarded to the wrong department, or otherwise mishandled. It is also important that the information collected by the Help Desk about a user be properly shared with other stakeholders in the business process.

The role of Help Desk tools is changing as well. The latest generation of Help Desk tools turn Web browsers into interactive clients of the help desk with the power to enter, query, and modify Help Desk requests. End users directly perform many of the services without assistance from the Help Desk staff.

Another key consideration of the Help Desk function in netcentric computing is that users must more effectively support themselves. In Internet environments, it is usually prohibitively expensive for a service provider to provide interactive Help Desk support to all interested Internet users. This is due to potential volumes of support requests as well as the diversity of technical environments that could be encountered. Consequently, it is often more reasonable to provide Internet users access to the tools required to support themselves. This can be accomplished through means such as a download site where patches, drivers, and self-help support materials are available.

Netcentric Help Desk organizations may also need to consider new metrics for measuring the performance of support personnel that consider interactions via e-mail or video. An example might be "number of e-mails answered per hour." In addition, existing metrics may need to be refined to fairly reflect netcentric characteristics.

The hours of Help Desk coverage may be affected by netcentric. To implement global 24x7 support, some service providers are deploying phased 8-hour "follow-the-sun" support windows that are based in different areas of the globe. The windows hand off support tickets among each other at the beginning and end of each phase so that all tickets are being addressed at all times. This is an effective way to ensure that there is no downtime for tickets; however, special attention should be paid to maintaining consistency of service across multiple regions when tickets are handed off.

In netcentric, there may be additional complexities of Help Desk operations introduced by global interactions. For example, multiple languages may need to be supported by Help Desks. Although this introduces people and process issues to Help Desk operations, from a tools perspective there may be a need for documentation to be published in multiple languages, and for the ability to switch easily between those versions.

OPERATIONS ARCHITECTURE INTEGRATION ISSUES

The operations architecture typically consists of different operations tools that focus on different functions, such as Help Desk or Fault Management. Each tool introduces a predetermined set of operations services such as core management logic and event generation. Although product selection decisions are often based on the functions that a product provides, true

integration of these tools into a cohesive operations architecture requires a service-based view rather than a functional view. In other words, operations architecture integration across functions is eased when different operations tools share core management logic, management data repositories, etc.

It is therefore important to consider the services provided by operations architecture tools when selecting operations tools.

These services are

- Core management logic
- Integration platform
- Event/data generation
- Event processing
- Repositories

Core Management Logic

Core Management Logic applies business roles to management data. Core Management Logic is typically specific to the function being served by an operations tool. For example, core management logic of a backup/restore application would initiate a backup process based on the time of day information it receives from a system clock. Core management logic receives data from event/data generation, event processing, and repositories services and then sends data for presentation or to repositories services. In addition, core management logic often polls the event/data generators for information.

Examples of Management Applications include a Help Desk package that automates the trouble ticketing process, a network management application such as HP OpenView, or a backup/restore utility that is used to create backups of server databases on a periodic basis.

Integration Platform

The integration platform provides a common platform for operations tools. At the lowest level this means deciding on common standards, interfaces, message formats, and file logging forms to be used with all the management tools. Although the Integration Platform can be homegrown, these applications are growing extremely complex, suggesting the use of one of many available third-party integration platforms.

There are two types of third party platforms available. The first group are framework-type products such as HP OpenView, CA-Unicenter TNG, and Tivoli Management Environment. These products are modular. Each module within the suite can be run separately; however, they all conform to

a common framework that allows for greater compatibility, integration, and better performance.

The second type of integration platform is point-solution oriented. Products such as Boole and Babbage implement this approach, which typically results in best-of-breed solutions for various management solutions, but a larger amount of integration work between tools is required.

Event/Data Generation

Event/data generation interacts with all the managed components in the execution and development environments to produce the required management information. The output of event/data generation services is actual raw management data that can then be processed and acted upon.

Event Processing

Event processing manipulates the raw data obtained by event/data generation services into a form on which operations personnel can take action. This service may perform several functions such as

- *Event filtering.* When management events are generated, event filtering mechanisms constantly compare predetermined event thresholds to current management events to determine the need for a management alert. If the threshold is exceeded, the event filtering function takes a specific action based on predetermined rules.
- *Alert generation.* When an event filter has noted the need for an alert, the alert generation function creates the proper notification. This may take one of several forms: a page, an e-mail, a display change (icon changes color to red), etc.
- *Event correlation.* Event correlation functions use logic to tie different events together with the intention of understanding potentials causes of problems. For example, nightly processing utilization shortages may be tied by event correlation functions back to a nightly batch job.
- *Event collection and logging.* It may be determined that historical analysis of management events is important. If so, the collection and logging of management events into repositories is important so that reporting and correlation activities can be performed at a future time.
- *Automated trouble ticket generation.* For certain events, it may be desirable for trouble tickets to be generated automatically in an organization's help desk system so that action can be taken.

Repositories

Repositories contain all the management data generated or used during the management process. These data include historical data, capacity

data, performance data, problem knowledge bases, asset databases, solution sets, and management information bases (MIBs).

MANAGING THE PHYSICAL ENVIRONMENT

The operations architecture can also include a set of tools and configurations used to ensure that the physical environment is manageable and protected against unplanned outages. The following are examples of systems that may serve this purpose.

- *Uninterruptible Power Supply (UPS).* To protect against loss of computing resources due to power outages, it is common for separate power units to be used as backup for critical computing resources. They may be implemented to support the power needs of an entire data center such as with an independent generator or may be implemented to support a single machine through battery-based UPS. Typically, UPS systems are designed to provide enough power for users and administrators to do the backups necessary to prevent catastrophic loss of information.
- *Raised floor.* To ease organization and provide spaces for systems components such as wiring and heating/cooling/power systems, it is common in data centers to implement raised floor environments.
- *Wiring/cabling.* The wiring and cabling of an operations architecture may be of many types. 10-BaseT (Unshielded Twisted Pair) and Fiber Optic cable are typically used for LAN and backbone links. Wiring systems are typically housed in wiring closets and/or under the raised floor to encourage an organized environment.

  Distribution panels for wiring to users' desktops may be used. Rather than having the cable from each desktop plug directly into a networking hub (which can cause difficulties in managing moves, adds, and changes) the distribution panel provides an intermediate connection between desktops and network hubs.
- *Fire suppression and climate control systems.* It is common in data centers or computer rooms to implement systems that protect computing resources from damage due to fire or other catastrophic events. Halon or other chemical systems may be implemented.
- *Disaster recovery.* If applicable, organizations may employ entirely redundant physical facilities to support disaster contingencies. In these cases, all of the above physical environment management services would be duplicated at the backup site.

MANAGING HARDWARE

In addition, there are hardware tools directly used in managing a distributed environment. These components are devoted to systems management functions. Examples of products that manage hardware include the following:

- *Management Servers.* These are servers that house management software. These may be a Tivoli Event Management Server, or a CA-Unicenter TNG server.
- *Management Consoles.* The operations center or computer room may have several consoles in a central location that serve as the nerve center for management information. On these consoles, management information for virtually all components of the environment can be accessed and action can be taken. System Administrators are typically the only users of management consoles.
- *Probes and sniffers.* These hardware devices provide diagnostic information by making physical measurements of electrical activity on wiring. These signals are then interpreted at a very low level (i.e., each data packet is dissected), which produces data that can be used to diagnose problems. Probes and sniffers can be extremely expensive, sometimes costing upward of $10,000 to $20,000.

CONCLUSION

The operations architecture consists of a set of tools that allows administrators to effectively manage a distributed environment. Although progress has been made in creating management frameworks that span the entire spectrum of management services, typical organizations must integrate different tools together to create a cohesive management picture. This integration effort is further complicated by the coexistence of host, client/server, and netcentric technologies.

Although this chapter has focused on operations tools, organization and processes are equally crucial success factors to managing a netcentric or client/server environment. As this chapter has shown, there are a multitude of considerations related to implementing and running a successful operations architecture; thus a structured, comprehensive framework that addresses the technology as well as people and process aspects is needed. Such a framework is introduced in the chapter in Section III that discusses the Management of Distributed Environments (MODE). Readers should understand the perspectives of both these chapters to ensure they have a comprehensive view of the operations architecture.